CODING OF AN ARBITRARILY SHAPED INTERLACED VIDEO IN MPEG-4

Xuemin Chen, R. Eifrig, A. Luthra and K. Panusopone

Advanced Technology Dept.,
Satellite Data Network systems,
General Instrument Corp., San Diego, CA 92121

ABSTRACT

The coding method in MPEG-4 for interlaced-video is introduced in this paper. Such a method is an extension of MPEG-2 interlaced coding tools to allow compression of arbitrarily-shaped interlaced-video. The new texture-and-shape coding tools discussed here for arbitrarily-shaped interlaced-video provide good compression performance. It is demonstrated here that these interlaced tools improve the coding efficiency of MPEG-4 video in both subjective and objective means.

1. INTRODUCTION

ISO/IEC 14496-2 [1] is the latest video compression standard developed by the ISO/IEC JTC1/SC29/WG11 working group. This standard, also known as MPEG-4, aims towards interactive multimedia services. It provides new compression tools such as shape and sprite coding, mesh and face object coding, etc.. To incorporate with a wide range of options for different multimrdia service, MPEG-4 divides itself into an array of profiles and levels. A profile is a defined subset of the entire bitstream syntax and a level is a defined set of constraints imposed on parameters in the bitstream. A combination of profile and level will specify a set of tools and a bound in the system performance.

A new coding method for interlaced-video is adopted in MPEG-4 main visual-object profile to provide an efficient solution for transmission and storage of both rectangular and arbitrarily-shaped interlaced video. To illustrate this method, a brief description of MPEG-4 coding tools for arbitrarily-shaped video is given in the next section. Then, major functional blocks (or tools) of this interlaced coding method are introduced and discussed in section 3. Next, simulation results for coding performance of this method are shown in section 4.

2. BRIEF DESCRIPTION OF CODING TOOLS FOR THE ARBITRARILY-SHAPED VIDEO

An arbitrarily-shaped video object(VO) appears in a certain area of the rectangular scene. In MPEG-4, such a video object is expressed by (1) the segmentation mask sequence which indicates pixel positions of the VO, (2) alpha-plane sequence which represents the blending contribution or the level of transparency of each pixel of the VO in the scene and (3) natural video which provides the texture content in the form of YUV 4:2:2.

The key concept behind MPEG-4 coding tools for the arbitrarily-shaped video is a systematic approach of coding alpha-plane togather with the texture blocks which contribute in the blending process. This systematic approach includes an unique shape coding and the traditional motion and texture coding process.

Fig.1 shows the simplified structure of MPEG-4 video coder for alpha plane and texture. The basic blocks of MPEG-4 video coder are similar to other standard hybrid DCT-Motion compensation(MC) video coder such as H.263 and MPEG-2. However, MPEG-4 texture coding aims several new algorithms such as AC/DC prediction, non-linear DC quantization, etc.. More modes, such as unrestricted MC, advanced prediction, direct mode, are also adopted in MC process to further improve coding efficiency. When integrating with shape coding, texture coding still processes all blocks as regular blocks. Block on the boundary of the object (part of the block have no blending contribution at all) will be converted into a full regular block by padding the content from the boundary. The extra information outside boundary are neglected in the decoder since the shape boundary is known.

There are two types of shape coding in MPEG-4 : binary alpha coding and grey scale alpha coding. In binary alpha plane, every pixel will have value either 0 (outside object) or some constant number such as 1 (inside object) and each square of 16x16 pixels in binary alpha plane, called binary alpha block (BAB), is formed such that each BAB aligns with its associated texture macroblock. Only a BAB that lies along the boundary is coded by binary shape coding using context based arithmetic encoding (CAE). CAE encodes the probability of the current binary alpha value which is predicted by the context. Context takes part of BAB in intra mode and the predicted BAB when motion estimation is used. Performance of CAE for shape sequence can be found in [2]. The grey-scale-alpha sequence is coded as the Y-component of natural video[3].

3. MPEG-4 INTERLACED CODING TOOLS

Efficiency of interlaced coding tools in MPEG-4 has been investigated under the core experiment P-14 of the coding efficiency adhoc group in MPEG video subgroup. Description of the core experiment is provided in [4]. Several tools specialized for interlaced video have been tried during the course of the experiments. All tools conducted in experiments involve texture coding and padding problems. A study of the interlaced shape coding tool has been done under the core experiment S-12 [8] of the coding of arbitrarily-shaped objects adhoc group in video subgroup for MPEG-4 video version 2. This section explains the interlaced coding tools that are currently adopted in MPEG-4 video version 1.

3.1. Interlaced video

Interlaced video is widely used in TV broadcasting. It aims to maximize the picture quality under tight bandwidth constraint. Instead of displaying the whole frame directly as in progressive format, two fields containing alternate lines constitute one video frame in interlaced format. Top field consists of all even lines while bottom field composes of all odd lines in the frame. Interlaced scan effectively double the frame rate by displaying each frame twice (one for top field and one for bottom field). Interlaced format is a compromise between two attributes of video:

- Frame rate, progressive format can remove every other frame and keep all lines in the remaining frame to achieve the same bandwidth as the interlaced video. Reduced in display rate results in flicker and motion related artifacts.
- Resolution, progressive format can maintain the frame rate and reduce the vertical resolution of each frame (only odd or even lines exist in all frames) leading to a limited spatial frequency and hence a blurring picture.

3.2. Adaptive frame/field DCT

MPEG-4 video uses DCT to exploit spatial correlation in macroblock. It is possible in interlaced video that a frame type macroblock has a smaller vertical correlation since adjacent lines come from different field e.g. when there is a motion in the scene. In this case, better decorrelation may be achieved in field type rather than frame type macroblock and field DCT should be used. When field DCT is performed, field type macroblock will be used for that macroblock in the coding process from DCT in the encoder until IDCT in the decoder. The selection of frame/field DCT is done on a macroblock basis when using interlaced coding tools. Chrominance blocks in 4:2:0 macroblock are not reordered in field DCT mode [4,7].

3.3. Adaptive frame/field motion compensation

Field motion compensation and interlaced direct mode are two major additions in interlaced motion compensation. Field motion compensation improves the coding efficiency because top and bottom field, scanning at a different instant, differ by motion of the object. When field motion compensation is used, field motion vectors are calculated for both top field and bottom field 16x8 macroblock. Although unrestricted motion compensation and advanced prediction are allowed, Overlapped block motion compensation (OBMC) is not applied in field motion compensation mode. Field motion vector has a half pixel accuracy in field coordinate i.e., full pixel vertical displacement corresponds to even integral value and a half pixel vertical is denoted by odd integral values. Field motion vector is coded using predictive coding based on motion vector predictor. Each component of the predictor is the median value of the candidate predictor vectors. These candidates consist of three motion vectors from three spatial adjacent blocks (for adjacent block that uses 8x8 motion vector), averages of 2 field macroblocks (for adjacent block that is field predicted) and/or macroblocks (for adjacent macroblock that uses 16x16 motion vector).

Interlaced B-VOP allows three following modes for motion compensation, field motion compensation, direct mode and 16x16 frame motion compensation. Interlaced direct mode can be used when the macroblock with the same coordinate of the future anchor VOP (co-located macroblock) uses field motion compensation. Direct mode is an algorithm for locating forward and backward motion vectors for the BVOP macroblock from the co-located future anchor VOP's motion vector. This calculation can be done for both progressive and interlaced prediction. Direct mode motion compensation determines forward and backward motion vector by adding the scaled forward motion vector of the co-located macroblock with the delta motion vector. Interlaced direct mode is an extension of progressive direct mode. For interlaced direct mode, 4 field motion vectors for both directions are determined based on 2 field motion vectors of the co-located macroblock. The generic operation of interlaced direct mode is shown in Fig. 2 and the relation among the motion vector is defined as follows

$$MV_{f,i} = (TR_{b,i} \times MV_i)/TR_{d,i} + MV_d \tag{1}$$

$$\text{if } MV_d = 0 \text{ then } MV_{b,i} = \frac{(TR_b - TR_d) \times MV_i}{TR_{d,i}}$$

$$else MV_{b,i} = (MV_{f,i} - MV_i) \tag{2}$$

$MV_{f,i}$ forward motion vector for field i, the reference field is the reference field of MV_i.

$MV_{b,i}$ backward motion vector for field i, the reference field is i.

MV_i field motion vector for field i of the co-located macroblock.

$TR_{b,i}$ temporal distance in fields between the past reference field for field i and field i of the current B-VOP.

$TR_{d,i}$ temporal distance in fields between the past reference field for field i and the future reference field for the current VOP's field i.

$$TR_{d,i} = 2 \times (TR_{future} - TR_{past}) + \delta \tag{3}$$

$$TR_{b,i} = 2 \times (TR_{current} - TR_{past}) + \delta \tag{4}$$

$TR_{future}, TR_{current}, TR_{past}$ are the frame number of the future, current and past frames in display order, and δ is -1,0,1 value determined from the structure of the field [1].

3.4. Alternate scan

Scanning is a mapping process which forms a 1-D data from a 2-D array of transform coefficient. 1-D data is subsequently coded by run-length and VLC. When using advanced prediction mode, MPEG-4 video uses adaptive scanning pattern (see [1] for detail) for all intra macroblocks based on DC prediction of their neighboring block and zigzag scan for all inter macroblocks. Efficiency of the VLC coding depends on a location of the nonzero coefficient along the scanning path. Adaptive scanning pattern may not yield the most benefit for interlaced video because the correlation in horizontal direction is usually dominated in interlaced video and the choice of adaptive scan may not follow this statistic. Upon selecting alternate scan mode in interlaced coding tools, MPEG-4 video will use alternate vertical scanning pattern for all macroblocks in that VOP [5,6].

3.5. Field padding

Padding provides the block based texture coding an interface to an arbitrarily shape object. It defines the content of the sample outside an arbitrarily shape object for motion prediction purpose. Padding is done via replicating and averaging the sample at the boundary of an object towards macroblock boundary. Padding process is performed twice, first horizontally and then vertically. The remaining area after horizontal and vertical padding is filled by extended padding. For interlaced video, a sample outside of a VOP replicates the value of the nearest boundary sample of the same field using field padding. Field padding includes the same repetitive and extended padding but the vertical padding of the luminance component is conducted separately for each field. By padding each field separately, there is no chance of replicating sample in one field to the other field thereby retaining correlation.

4. PERFORMANCE OF INTERLACED CODING TOOLS

This section provides the simulation results of interlaced coding tools. The first experiment uses interlaced coding tools without shape coding (rectangular object) at fixed quantized level (QP=12). Test sequence 'Stefan' (720x480 pixels, 300 frames) is used in this test. The results show the comparison between MPEG-4 with and without interlaced tools together with MPEG-2 for luminance PSNR, total bits, DCT bits, Motion vector bits, Proportion of Field DCT and Field MC. Comparison also be made between MPEG-2 and H.263 quantization schemes. Constant bitrate simulations are performed with the MPEG-2 test model 5 (TM5) rate control algorithm. The simulation picture structure is IBBP... coding at bitrate 3 Mbps using H.263 quantization method.

Table 1. Performance at constant bitrate

Sequence	M4p	M4i	MPEG-2
Fun Fair	29.42	29.84	28.23
Football	32.15	33.00	31.62
Stefan	28.91	29.79	28.26

Performance of the individual tool is given in Tables 2-7. A pair of tables are used to explain the efficiency of each tool. Tables 2,4, and 6 show average bits decrease due to Field MC, Field DCT (B-VOP) and direct mode, respectively. Table 3,5, and 7 show the PSNR improvement of the same set of tools. The performance gain comes from the use of interlaced coding tool. All of the data obtained in these tables using the H.263 quantization method. Table 8 shows the improvement from alternate scan for 'Stefan' sequence.

Table 2. Field Motion Compensation

Average bits per VOP decrease				
Q	Type	Fair	Football	Stefan
8	M4p	230732	163467	222764
8	M4p+fldMC	220403	140826	183291
8	pct. improve	4.47	13.6	17.7
12	M4p	145613	106029	143813
12	M4p+fldMC	138912	94098	116210
12	pct. improve	4.6	11.3	19.2
16	M4p	105994	79449	104514
16	M4p+fldMC	101333	72874	84158
16	pct. improve	4.4	8.3	19.4

Table 3. Field Motion Compensation

PSNR improvement				
Q	Type	Fair	Football	Stefan
8	M4p	32.80	34.51	33.38
8	M4p+fldMC	32.89	34.79	33.58
8	increase	0.09	0.28	0.20
12	M4p	30.48	32.40	30.85
12	M4p+fldMC	30.62	32.79	31.19
12	increase	0.14	0.39	0.34
16	M4p	28.94	31.01	29.10
16	M4p+fldMC	29.12	31.49	29.57
16	increase	0.18	0.48	0.47

Table 4. Field DCT (IBBP...)

Average bits per VOP decrease				
Q	Type	Fair	Football	Stefan
8	M4i-fldDCT	196224	128355	156781
8	M4i	176379	116800	150177
8	pct. improve	10.1	9.0	4.2
12	M4i-fldDCT	132447	91872	107934
12	M4i	119081	84720	103196
12	pct. improve	10.1	7.8	4.4
16	M4i-fldDCT	103203	75636	84974
16	M4i	93876	70685	81510
16	pct. improve	9.0	6.5	4.1

Table 5. Field DCT (IBBP...)

PSNR improvement				
Q	Type	Fair	Football	Stefan
8	M4i-fldDCT	32.04	33.99	32.76
8	M4i	32.16	34.16	32.79
8	increase	0.12	0.17	0.03
12	M4i-fldDCT	29.97	32.23	30.62
12	M4i	30.10	32.41	30.66
12	increase	0.13	0.18	0.04
16	M4i-fldDCT	28.62	31.13	29.16
16	M4i	28.75	31.31	29.19
16	increase	0.13	0.18	0.03

Table 6. Direct Mode

Average bits per VOP decrease				
Q	Type	Fair	Football	Stefan
8	M4i-fldDCT	112635	84553	99690
8	M4i	110939	84055	99357
8	pct. improve	1.5	0.6	0.3
12	M4i-fldDCT	79283	65741	70430
12	M4i	77271	65045	69753
12	pct. improve	2.5	1.1	1.0
16	M4i-fldDCT	66370	58191	58546
16	M4i	64159	57432	57664
16	pct. improve	3.3	1.3	1.5

Table 7. Direct Mode

PSNR improvement				
Q	Type	Fair	Football	Stefan
8	M4i-fldDCT	31.59	33.67	32.21
8	M4i	31.58	33.64	32.19
8	increase	-0.01	-0.03	-0.02
12	M4i-fldDCT	29.67	32.05	30.24
12	M4i	29.66	32.03	30.22
12	increase	-0.01	-0.02	-0.02
16	M4i-fldDCT	28.42	31.04	28.90
16	M4i	28.41	31.02	28.87
16	increase	-0.01	-0.02	-0.03

Table 8. Alternate scan

Q	Type	Pct. improvement
8	III...	9.23
10	III...	9.68
12	III...	9.95
8	IPP...	8.43
10	IPP...	8.44
12	IPP...	8.34

5. CONCLUSIONS

MPEG-4 offers an efficient compression tool for an arbitrarily shaped interlaced video. Interlaced coding tools in MPEG-4 version 1 (adaptive frame/field DCT, adaptive frame/field motion compensation, alternate scan, field padding) improves the coding efficiency of the texture coding for interlaced video. It is obvious from the simulation that interlaced shape coding tools improves the coding efficiency for both rectangular and arbitrarily shaped object. Our results also show that MPEG-4 with interlaced coding tools offers a higher PSNR than MPEG-2 at the same bit rate.

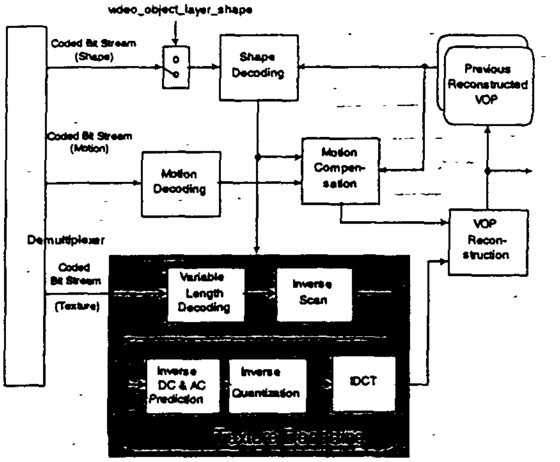

Figure 1: Block diagram of binary shape decoding

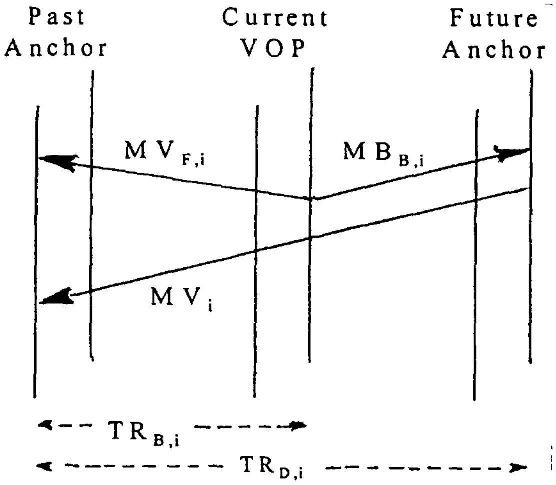

Figure 2: Direct mode of interlaced video

6. REFERENCES

[1] ISO/IEC 14496-2, Information Technology - Coding of Audio-Visual Objects : Visual International Organization for Standardisation, Mar. 1998, Committee Draft (MPEG-4 Video)

[2] N. Brady, F. Bossen and N. Murphy, Context-based Arithmetic Encoding of 2D Shape Sequences, ICIP, 1997.

[3] W. Chen, M. Lee, Alpha-Channel Compression in Video Coding, ICIP, 1997.

[4] B. Eifrig, X. Chen, A. Luthra, Updated Description of Interlaced Video Coding, International Organization for Standardisation, Bristol, GB, Apr. 1997, ISO/IEC JTC1/SC29/WG11 MPEG97/2123.

[5] K. Panusopone, X. Chen, Efficiency of Alternative Vertical Scan for VLC of Interlaced video, International Organization for Standardisation, Fribourg, CH, Oct. 1997, ISO/IEC JTC1/SC29/WG11 MPEG97/2640.

[6] T. Nishi, C. S. Boon, S. Kadono, Alternative Vertical Scan Mode for Interlaced Video, International Organization for Standardisation, Fribourg, CH, Oct. 1997, ISO/IEC JTC1/SC29/WG11 MPEG97/2671.

[7] B. Eifrig, X. Chen, A. Luthra, Interlaced Video Coding Results (Core Exp P-14), International Organization for Standardisation, Bristol, GB, Apr. 1997, ISO/IEC JTC1/SC29/WG11 MPEG97/2007.

[8] J. S. Shin, J. W. Chung and S. Han, Description of CE S12 (Interlaced Shape & Texture Coding), International Organization for Standardisation, Dublin, IE, Jul. 1998, ISO/IEC JTC1/SC29/WG11 MPEG98/3765.